



AF
Iw

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:

Slaughter, et al.

Serial No. 09/653,241

Filed: August 31, 2000

For: MECHANISM AND
APPARATUS FOR SECURITY
OF NEWLY SPAWNED
REPOSITORY SPACES IN A
DISTRIBUTED COMPUTING
ENVIRONMENT

§ Group Art Unit: 2143
§
§ Examiner: Lezak, Arrienne M.
§
§ Atty. Dkt. No.: 5181-67100
§ P4978

**CERTIFICATE OF MAILING
37 C.F.R. § 1.8**

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

Robert C. Kowert
Name of Registered Representative

January 26, 2005
Date

[Signature]
Signature

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed November 26, 2004, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 011070/0145, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-33 stand finally rejected. The rejection of claims 1-33 is being appealed. A copy of claims 1-33 is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Intelligent devices are becoming more common and the desire to improve networking capabilities is ever increasing. Traditional networks are complex to set up, expand and manage. For example, adding hardware or software to a network often requires a network administrator to load drivers and configure systems. Various technologies exist for improving the addition of devices to a network. However, these solutions are often limited to specific peripheral buses and are not suitable for general networks. A more recent technology, Jini from Sun Microsystems, Inc., seeks to simplify the connection and sharing of devices such as printers and disk drives on a network. Jini

allows for distributed computing where the capabilities of the various devices are shared on a network. Jini requires that each Jini enabled device has a certain amount of memory and processing power.

Object based distributed computing systems often need persistent storage. However, attempts at object storage are often language and operating system specific. In addition, object storage systems are generally too complicated to be used with many small, embedded systems. For example, the Jini technology uses JavaSpaces as persistent object containers. JavaSpaces from Sun Microsystems, Inc., draws from the parallel processing work of David Gelernter, a computer science professor at Yale University. Gelernter's set of functions named "Linda" create a shared memory space called a TupleSpace, in which results of a computer's processes or the processes themselves may be stored for access by multiple CPUs. Linda therefore provides a global shared memory for multiple processors.

Another technology that extends Linda is T Spaces from IBM Corporation. T Spaces extends the basic Linda TupleSpace framework with real data management and the ability to download new datatypes and new semantic functionality. T Spaces provides a set of network communication buffers and a set of APIs for accessing those buffers. Like many solutions, T Spaces uses a code-centric programming model and shares the drawbacks of such a model. Additionally, T Spaces is implemented in the Java programming language and therefore requires a Java Virtual Machine or other means of executing Java bytecode, such as a Java-capable microprocessor.

Independent claim 1 is directed to a method that employs schema-defined messaging in a distributed computing environment for securely spawning a new network addressable space service from an existing network addressable space service. The method includes accessing a first space that has a network-addressable storage location. Information usable to access the first space is provided in an advertisement for the space. For instance, as described on page 13 of the Specification, a distributed computing environment may rely on "spaces" or object repositories to provide a rendezvous

mechanism or catalyst for the interaction between clients and services. Service providers may advertise services in a space. Clients may find the advertisements in a space and use the information from an advertisement to access a service using a messaging mechanism of the distributed computing environment where messages are defined according to a schema for the space. Many spaces may exist, each containing advertisements that describe services or content. Thus, a space may be a repository of advertisements of services and/or data, which may be raw data or advertisements for data, such as results. *See e.g.* FIGs. 6, 8, 11b, 13, 15, 18, 19, 20, 23, 24, 27, 28, 30, 29, 31, 32A, 32B, 38, 41, 42; page 14, lines 8-24; page 30, lines 8-14; page 33, lines 9-18.

The advertisement for a space may comprise a schema that specifies messages to invoke functions of the space. Like any service, a space may have an advertisement, which a client of the space obtains in order to be able to run that space service. A space's own advertisement may include a schema (such as an XML schema), a credential or credentials, and a URI (Uniform Resource Identifier) or other address indicating how to access the space. A client of a space may run various space facilities by sending messages to the space service. The schema may specify a set of messages that clients of the service may send to the service to invoke functionality of the service. The schema may define the client-service interface. Together, the URI and the schema in an advertisement may indicate how to address and access the service. *See e.g.* page 14, lines 8-24; page 33, lines 9-18; page 34, line 22 – page 35, line 2.

A client may request creation of a second space by sending one of the messages specified in the schema (from the advertisement for the first space) to the first space and the second space may be created in response to the client request. For example, a second space service with a second space may be created, at a second network address. Storage for the newly spawned space may be allocated using the same facility used by the original space for storage. Upon its creation, the second space may be operable to store a set of information according to the same storage model as the first space. Also, a spawned space may share a common service facility with its original (or parent) space.

See e.g. FIGs. 41, 42; page 17, line 21 – page 18, line 3; page 18, line 29 – page 19, line 12; page 85, line 7-18; page 85, line 20 – page 86, line 17.

Similarly to the first space, the second space may include a network-addressable storage location, and information usable to access the second space may be provided in an advertisement for the second space. The advertisement for the second space may comprise a second schema that specifies messages to invoke functions of the second space. *See e.g.* FIGs. 41, 42; page 18, line 20-27; page 85, line 20 – page 86, line 17; page 87; lines 4-10; page 113, lines 15- 26. The second space may be initially configured to only permit access to the requesting client. For example, an authentication service associated with the second space may be initialized, whereby the second space may be configured to permit access only to a client holding a root authentication token. The root authentication token may be sent to the requesting client or service. *See e.g.* FIG. 42; page 19, lines 14-23; page 86, lines 19-25; page 113, line 28 – page 114, line 19.

The client may access the second space by sending one of the messages in the second schema (from the advertisement for the second space) to the second space. For instance, a requesting client may access the second space by sending to the second space at least one of the messages specified in the second schema and by using the root authentication token. *See e.g.* FIG 41; page 18, lines 26-27; page 75, lines 1 – 17; page 86, lines 10-17.

Independent claim 12 is directed to a system including a first space communicable coupled to a client. The first space includes a network-addressable storage location, and information usable to access the first space is provided in an advertisement for the first space. The client recited in claim 12 is operable to access the first space and request the creation of a second space in a similar manner as described above in regard to independent claim 1. Additionally, the client recited in claim 12 is also operable to access the second space in a similar manner as described above in regard to independent claim 1. Please refer to the description of independent claim 1 above for examples regarding such spaces and advertisements.

Independent claim 23 is directed to a carrier medium comprising program instructions computer executable to implement a method similar to that recited in independent claim 1. Please refer to the summary and description of claim 1 for a summary of examples of such a method.

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 12 and 23 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Ford et al. (U.S. Patent 5,963,947) (hereinafter “Ford”). Please note that in the Final Action, the Examiner refers to Ford et al. (U.S. Patent 5,963,947) as “Lehman (‘947).” Appellants, however, refer to this patent as “Ford” for clarity over Lehman et al. (U.S. Patent 5,974,420).

2. Claims 1, 5-12, 16-23 and 27-33 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Lehman (U.S. Patent 5,974,420) (hereinafter “Lehman”) in view of Ford.

3. Claims 2-4, 13-15 and 24-26 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Lehman in view of Ford in further view of Oliver (U.S. Publication 2002/0133412).

4. Claims 1-33 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Lehman in view of Ford and in further view of “IBM Systems Journal, Vol. 37, No. 3” by Wyckoff, McLaughry, Lehman and Ford, 1998 (hereinafter “Wyckoff”).

VII. ARGUMENT

First Ground of Rejection:

Claims 1, 12 and 23 are rejected under 35 U.S.C. § 102(e) as being anticipated by Ford et al. (U.S. Patent 5,963,947) (hereinafter “Ford”). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

In the Advisory Action of November 17, 2004, the Examiner states:

Examiner’s understanding of the pending application is that applicant has a means that provides the particular advantage of lowering memory footprint and computing resources while maintaining the ability to find and invoke services dynamically. Thus the implementation of the pending application attempts to distinguish itself from prior art (e.g. Jini and JavaSpaces), by providing a non-obvious advantage (Specification: p. 3, lns. 10-30 and p. 12, lns. 6-15). However, this advantage is currently addressed in the specification rather than in the claims themselves.

Appellants remind that Examiner that the invention is defined by the claims. Examination should be based on the language of the claims, not on the Examiner’s guess as to what the invention might be based on her reading of the specification. Appellants note that numerous novel features are disclosed in the specification. The claims of the present application are generally directed to securely spawning a new network-addressable space service from an existing network-addressable space service using schema-defined messaging in a distributed computing environment.

In the Advisory Action, the Examiner further states:

The claims as currently written appear to describe more of the ***kind of problem*** that applicant’s invention is to address, rather than the ***specific means*** of how the applicant’s invention solves this problem. Thus the claims as currently written not only cover the actual invention of applicant, but also all other means to solve the kind of problem described. This includes the T-Spaces of Ford, Lehman, as well as others. Thus if the claims were to be amended to address the specific implementation of the applicant’s invention, it would better distinguish itself from the prior art.

Contrary to the Examiner’s assertion, the claims as currently written do not merely recite a “kind of problem.” Instead, the claims recite specific limitations of a method, a system

and a computer medium. For instance, independent claim 1 recites, in part, accessing a first space, a requesting client requesting creation of a second space, creating the second space in response to the requesting client requesting creation, and the requesting client accessing the second space. Additionally, claim 1 recites specific limitations such as wherein the first space comprises a first network-addressable storage location, wherein information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space. As shown in more detail below, these specific limitations (as well as other limitations of the claims) are not taught or suggested by the evidence of record.

Claim 1:

Regarding claim 1, contrary to the Examiner's assertion, Ford fails to teach that information usable to access a first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space. Ford teaches a method, called "T Spaces" including dynamically adding functionality to a server by allowing new operators and JAVA-based operator handlers to be installed on a server for future use (Ford, column 7, line 66 – column 8, line 17, and column 8, lines 26-35). Ford clearly does not disclose an *advertisement for the first space* that provides information usable to access the first space and that *comprises a schema* that specifies one or more messages usable to invoke functions of the first space. In contrast, Ford teaches that T-Spaces clients include a Tuplespace class and a communications library to communicate with a T-Spaces server (Ford, column 6, lines 29-33). Thus, rather than using a schema from an advertisement for a space, Ford teaches that clients use a building communications library that includes functions for communicating with a T-Spaces server. Nowhere does Ford teach that such a communication library uses a *message schema* from an advertisement for a space. Nor would there be any need for such a schema in Ford's system because every T Spaces

client already has a communication library that includes programmatic functions for communicating with a T-Spaces server.

In the Response to Arguments section of the Final Office Action the Examiner includes a discussion regarding two different types of schemas. However, both of the types of schemas referred to by the Examiner are schemas that define data structures (one type of schema for a data repository that describes data tables and their relationships and another type of schema that describes the data layout of a particular database record). Neither of these types of schemas have anything to do with a *message schema specifying messages* usable to invoke functions of a space, as recited in Appellants' claimed invention. Nor are the Examiner's schemas described by Ford. Ford's Tuplespace class and communication library do not involve messages usable to invoke functions on a space that are *specified in a schema in an advertisement* for the space. Not only does Ford fail to mention anything regarding such a message schema, Ford also fails to disclose an advertisement comprising such a message schema and providing information usable to access the space.

Also in the Response to Arguments of the Final Office Action, the Examiner argues that Ford teaches advertisements as recited in claim 1. Specifically, the Examiner argues that T Spaces stores data to be read by different processes (or clients) and exposes stored information programmatically. The Examiner then asserts (erroneously) that Appellants' Specification describes the use of an advertisement as programmatically exposing data stored in a repository. *See* Final Office Action, page 9, paragraphs 20-21. However, the Examiner has misquoted and misinterpreted the Appellants' Specification.

The Examiner cites page 13, lines 10-15 of the Specification. This portion of the Specification describes how service providers may advertise services in spaces and how a client may locate the advertisements in a space and use information from an advertisement to access a service. Thus, the Specification does not describe advertising as merely "the storing of data in a repository" as the Examiner states. Just because advertisements may be stored in a data repository does not imply that any data stored in a

repository constitutes an advertisement. For example, another passage of the Specification cited by the Examiner (page 13, line 15-20) specifically describes how an advertisement for a space may include an XML schema, one or more credentials, and a URI. Yet, the Examiner asserts that this passage in conjunction with the aforementioned passage describes service advertisements as nothing more than data stored in a repository that is programmatically exposed. Thus, the Examiner has not only misinterpreted the Specification, but is also attempting to ignore the specific language of the claim based on her erroneous interpretation of the Specification.

The Examiner is ignoring specific limitations from claim 1. It is clearly improper to reject a claim based upon selected passages from the Specification, especially when incorrectly interpreted, while ignoring the specific limitations and language of the claim. For example, as stated above, claim 1 recites, in part, that information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space *comprises a first schema*, and wherein the first schema *specifies one or more messages* usable to invoke functions of the first space. Thus, the Examiner is attempting to reject claim 1 based upon her misinterpretation of selected passages from the Specification and the prior art while not taking into account the specific language of claim 1.

The Examiner has completely failed, both in the rejection of claim 1 and in the Response to Arguments section, to ever cite any portion of Ford describing an advertisement for a first space that comprises a first schema specifying messages usable to invoke functions of the first space.

Further regarding claim 1, contrary to the Examiner's assertion, Ford clearly fails to teach a *client requesting creation of a second space by sending to the first space one of the messages specified by the first schema*. As noted above, Ford does not disclose a schema specifying messages usable to invoke functions of a space. Instead of teaching that a client creates a new space by sending one of the messages specified by a schema, Ford teaches the use of a specific operator, `NewTupleSpace()` that originates as a method

invocation on T Spaces client (Ford, column 7, lines 7-10 and lines 21-30). The Examiner cites a passage of Ford (column 9, lines 34-43) that only describes how an operator (a command) is received from a client and that new functionality is added in response to receiving the operator. However, the cited passage fails to disclose that receiving such an operator involves the *client sending to the first space one of the messages specified in a schema provided in an advertisement that provides information usable to access the first space*. Thus, Ford does not teach requesting the creation of a new space by sending a message specified in a schema, but rather that Ford describes invoking methods of a Tuplespace class linked into a T Spaces client. Nowhere does Ford teach that a T Spaces client, a Tuplespace class, or a T Spaces communication library sends to a space a message specified in a schema to request creation of a new space.

Additionally, Ford does not teach wherein the second space is initially configured to permit access only to the requesting client. Instead, Ford teaches only that “[u]sers can establish security policies by setting user and group permissions on a Tuplespace basis” (Ford, column 5, lines 10-12). Ford also teaches that indications of a client’s access control privileges may be included as method parameters (Ford, column 7, lines 31-35), and that “only designated entities should have access control privileges to add new factories and handlers” (Ford, column 8, lines 60-63). Further, the Examiner’s cited passage (Ford, column 9, lines 34-46) only mentions that new functionality may be added to a Tuplespace without mentioning anything regarding the initial access control privileges regarding any newly added functionality. Thus, Ford clearly fails to teach *wherein the second space is initially configured to permit access only to the requesting client*. Appellants note that the Examiner has failed to present any rebuttal of this argument.

Furthermore, Ford does not teach that information usable to access the second space *is provided in an advertisement for the second space*, wherein the *advertisement for the second space comprises a second schema*, and wherein the *second schema specifies one or more messages usable to invoke functions* of the second space. Similar to the

discussion above regarding an advertisement for a first space, the Examiner cites column 9, lines 43-46 that does not mention anything about an advertisement for the second (created) space providing information usable to access the second space and that comprises a second schema that specifies messages usable to invoke functions of the second space. In fact, the cited passage only refers to the addition of new functionality in T-spaces, without describing anything regarding accessing the new functionality after it has been created.

Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Ford clearly does not anticipate Appellants' claim 1 for at least the reasons presented above.

Claim 12:

Regarding claim 12, contrary to the Examiner's assertion, Ford fails to teach that information usable to access a first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space.

As described above regarding the rejection of claim 1, Ford teaches a method, called "T Spaces" for dynamically adding functionality to a server that allows new operators and JAVA-based operator handlers to be installed on a server for future use (Ford, column 7, line 66 – column 8, line 17, and column 8, lines 26-35). Ford does not disclose an advertisement for the first space that provides information usable to access the first space and that comprises a schema that specifies one or more messages usable to invoke functions of the first space.

Ford also fails to anticipate a client operable to request creation of a second space *by sending to the first space one of the messages specified by the first schema*. As noted above, Ford does not disclose any sort of schema that specifies messages usable to invoke functions of a space. Instead, Ford teaches the use of a specific operator, NewTupleSpace(). Thus, Ford does not teach requesting the creation of a new space through the sending a message specified in a schema, but rather that Ford describes method invocations of TupleSpace class in a T-Spaces client. Nowhere does Ford teach that a T-Spaces client, a TupleSpace class, or T-Spaces communication library would send a message specified in a schema to request creation of a new space.

Additionally, Ford does not teach wherein the second space is initially configured to permit access only to the requesting client. Instead, Ford teaches only that “[u]sers can establish security policies by setting user and group permissions on a TupleSpace basis” (Ford, column 5, lines 10-12), that indications of a client’s access control privileges may be included as method parameters (Ford, column 7, lines 31-35), and that “only designated entities should have access control privileges to add new factories and handlers” (Ford, column 8, lines 60-63). Thus, Ford clearly fails to teach *wherein the second space is initially configured to permit access only to the requesting client*.

Furthermore, Ford does not teach wherein information usable to access the second space *is provided in an advertisement for the second space*, wherein the *advertisement for the second space comprises a second schema*, and wherein the second schema specifies one or more messages usable to invoke functions of the second space. Similar to the discussion above regarding an advertisement for a first space, the Examiner cites column 9, lines 43-46 that does not mention anything about an advertisement for the second (created) space providing information usable to access the second space and that comprises a second schema that specifies messages usable to invoke functions of the second space.

In general, the arguments presented above regarding the rejection of claim 1 apply to the rejection of claim 12 as well.

Claim 23:

Regarding claim 23, contrary to the Examiner's assertion, Ford fails to teach program instructions that are computer-executable to implement wherein information usable to access a first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space. As noted above, Ford teaches a method, called "T Spaces" for dynamically adding functionality to a server that allows new operators and JAVA-based operator handlers to be installed on a server for future use (Ford, column 7, line 66 – column 8, line 17, and column 8, lines 26-35). Ford does not disclose an advertisement for the first space that provides information usable to access the first space and that comprises a schema that specifies one or more messages usable to invoke functions of the first space. Instead, Ford teaches that T-Spaces clients include a Tuplespace class and a communications library to communicate with a T-Spaces server (Ford, column 6, lines 29-33). Please refer to the remarks above regarding the rejections of claims 1 and 12 for a more detailed discussion regarding Ford's failure to anticipate wherein information usable to access a first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space.

Further regarding claim 23, Ford also fails to anticipate a client requesting creation of a second space *by sending to the first space one of the messages specified by the first schema*. Instead, Ford teaches the use of a specific operator, NewTupleSpace(), and that such commands originate as a method invocation on T-Spaces client (Ford, column 7, lines 7-10 and lines 21-30). Please note that the arguments outlined above regarding the rejections of claims 1 and 12 also generally apply to claim 23.

Additionally, Ford does not teach wherein the second space is initially configured to permit access only to the requesting client. Instead, Ford teaches only that “[u]sers can establish security policies by setting user and group permissions on a Tuplespace basis” (Ford, column 5, lines 10-12), that indications of a client’s access control privileges may be included as method parameters (Ford, column 7, lines 31-35), and that “only designated entities should have access control privileges to add new factories and handlers” (Ford, column 8, lines 60-63).

Furthermore, Ford does not teach wherein information usable to access the second space *is provided in an advertisement for the second space*, wherein the *advertisement for the second space* comprises a second schema, and wherein the second schema specifies one or more messages usable to invoke functions of the second space. Similar to the discussion above regarding an advertisement for a first space, the Examiner cites column 9, lines 43-46 that does not mention anything about an advertisement for the second (created) space providing information usable to access the second space and that comprises a second schema that specifies messages usable to invoke functions of the second space. In fact, the cited passage only refers to the addition of new functionality in T-spaces, without describing anything regarding accessing the new functionality after it has been created.

As noted above, the arguments and remarks presented above regarding the rejections of claims 1 and 12 also generally apply to the rejection of claim 23.

Second Ground of Rejection:

Claims 1, 5-12, 16-23 and 27-33 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Lehman (U.S. Patent 5,974,420) (hereinafter “Lehman”) in view of Ford. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Lehman describes T Spaces in an almost identical manner as Ford, but additionally teaches the use of a Rhonda operator for use within T Spaces wherein two

Rhonda operators swap their tuples when certain arguments match (Lehman, column 8, line 65 – column 9, line 3). Thus, the arguments made above in regard to the first ground of rejection generally apply to this ground of rejection as well.

Claims 1, 6 and 11:

Please note that the arguments presented above against the § 102(e) rejection of claim 1 regarding Ford apply to this rejection of claim 1 as well. Furthermore, Lehman provides essentially the same discussion of T Spaces as Ford and thus Lehman fails to overcome any of the deficiencies of Ford noted above.

Further regarding claim 1, contrary to the Examiner's contention, Lehman in view of Ford fails to teach or suggest accessing a first space, wherein information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space. Since Lehman includes an almost identity discussion of T Spaces as Ford, Lehman fails to overcome any of the deficiencies of Ford's teachings, as described above regarding the § 102(e) rejection of claim 1.

Specifically, Lehman in view of Ford fails to teach or suggest that information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space. Lehman, like Ford, teaches that T Spaces clients include a Tuplespace class and a communications library that includes methods to communicate with a T Spaces server (Lehman, column 4, lines 55-61). Thus, a T Spaces client in Lehman clearly does not use a schema that specifies messages usable to invoke functions of a space.

Also, Lehman in view of Ford clearly fails to teach or suggest a client requesting creation of a second space *by sending to the first space one of the messages specified by*

the first schema. Lehman, like Ford, teaches the use of a specific operator, NewTupleSpace(), and that such commands originate as a method invocation on T Spaces client (Lehman, column 5, lines 50-60). The examiner cites passages of Lehman that describe Lehman's Rhonda Operator that is used when two T Spaces clients exchange information via the T Spaces server (column 8, line 61-column 9, line 8, and column 9, lines 35-48) and that include broad statements regarding how "different computer programming languages, database systems, operating environments, and operating systems" could be substituted for those described by Lehman. Neither of the Examiner's cited passages refers to creating a new space by sending to a space a message specified by a schema.

Additionally, Lehman in view of Ford, does not teach or suggest that the second space is initially configured to permit access only to the requesting client. Nor does they teach or suggest that information usable to access the second space is provided in an advertisement for the second space, wherein the *advertisement for the second space comprises a second schema*, and wherein the second schema specifies one or more messages usable to invoke functions of the second space. As Lehman teaches essentially the identical T Spaces system as Ford, the arguments above regarding the § 102(e) rejection of claim 1 as applied to Ford, apply to Lehman as well.

Furthermore the discussion above regarding the Examiner's comments in the Response to Arguments section regarding Ford's teachings, further apply to Lehman's teachings as well.

Claim 5:

Appellants note that the rejection of claim 5 is improper because claim 5 depends from claim 4 and the Examiner has not rejected claim 4 as being unpatentable over Ford in view of Lehman. As discussed below under the Third Ground of Rejection, the Examiner admits that Ford in view of Lehman fails to teach the limitations of claim 4 and

relies upon a third prior art reference (Oliver) for the rejection of claim 4. Claim 5, since it is dependent from claim 4, also includes the limitations recited in claim 4.

Further regarding claim 5, Lehman in view of Ford fails to disclose or suggest a second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service. The Examiner does not cite any particular passage of Lehman, but instead states, “[t]o add an aspect of optional functionalities to [Ford] would have been obvious ... as T Spaces provides a powerful mechanism for inter-process communication and synchronization.” This is clearly pure hindsight speculation by the Examiner.

Lehman’s extended functionality is described as providing a mechanism for exchanging information between two or more processes, providing synchronous, anonymous rendezvous and data exchange, providing a mechanism for controlling multiple processes, and providing barrier synchronization among a dynamically defined group of processes (Lehman, column 3, lines 21-30). Thus, while Lehman teaches that his Rhonda Operator can provide various means of inter-process communication, nowhere does Lehman describe or suggest anything regarding a second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service.

Appellants note that the Examiner has never provided any rebuttal of this argument.

Claim 7:

Regarding claim 7, Lehman in view of Ford fails to teach or suggest that the first schema is expressed in a data representation language; and wherein the second schema is expressed in the data representation language.

The Examiner rejected Claim 7 “on the basis that Lehman teaches the use of different computer languages.” At the Examiner’s cited reference (Lehman, column 9, lines 10-14), Lehman states, “different computer *programming* languages, database systems, operating environments, and operating systems could be substituted for those described herein” (emphasis added). Appellants assert that Lehman is referring to programming languages that may used to create T-Spaces clients, servers and other programs, such as, Java, C, C++, and the like. In other words, Lehman is describing how different programming languages may be used to create programs that can utilize his system. A general reference to programming languages does not suggest the use of a data representation language, which is a specific type of language for representing or describing data. Programming languages such as mention in Lehman have no bearing on a *message schema* that is expressed in a *data representation* language and that specifies messages usable to invoke functions of a space.

The Examiner has responded to Appellants’ argument above by pointing out the differences between a Data Manipulation Language and a Data Definition Language (Final Office Action, pages 11-12, paragraph 25). Appellants submit however that the Examiner has misunderstood Appellants’ argument. The issue is not whether a data representation language may be both statically stored or dynamically created; instead, the issue is whether or not Lehman’s general statement regarding *programming* languages suggests the use of a *data representation* language for a schema specifying messages usable to invoke functions of a space. Clearly, it does not.

Claim 8:

Regarding claim 8, Lehman in view of Ford fails to teach or suggest wherein the first schema is expressed in a data representation language; wherein the second schema is expressed in the data representation language; and wherein the data representation language comprises eXtensible Markup Language (XML). As with the rejection of claim 7 discussed above, the Examiner has rejected claim 8 “on the basis that Lehman teaches

the use of different computer languages.” The Examiner admits that neither Ford nor Lehman teaches discloses the use of XML with T-Spaces.

However, the Examiner states that it would have been obvious to combine XML with T-Spaces and states that the motivation to combine XML with T-Spaces is noted within Lehman, “wherein the use of different computer languages is recognized as possible” and the Examiner further states that “as XML was in existence at the time of invention ... the use of the same as a programming language would have been obvious.” As discussed above regarding the rejection of claim 7, the Examiner’s cited reference states that “different computer *programming* languages ... could be substituted for those described” (Lehman, column 9, lines 10-14). Lehman is referring to *programming* languages that may used to create programs that may utilize his invention and thus has no bearing on data representation languages, such as XML, which are used for very different purposes than computer programming languages. Furthermore, the Examiner’s discussion regarding the traditional uses of XML is irrelevant. In the prior art, XML schemas have been used to describe data structures and content, not to specify messages usable to invoke functions of a space.

Neither Lehman nor Ford teaches anything that suggests the use of a data representation language. Therefore, the combination of Lehman in view of Ford fails to teach or suggest any desire or benefit to using a data representation language, as the Examiner contends. Furthermore, the mere fact that XML was in existence at the time of Appellants’ invention does render its use obvious as implied by the Examiner. As the Federal Circuit stated in *In re Kotzab*, 55 USPQ2d 1313, 1316 (Fed. Cir. 2000):

Most if not all inventions arise from a combination of old elements. However, identification in the prior art of each individual part claimed is insufficient to defeat patentability of the whole claimed invention. Rather, to establish obviousness based on a combination of the elements disclosed in the prior art, there must be some motivation, suggestion or teaching of the desirability of making the specific combination that was made by the applicant.

Additionally, Appellants arguments above regarding the rejection of claim 7 also apply to the rejection of claim 8.

Claim 9:

Regarding claim 9, Lehman in view of Ford does not teach or suggest reading a service advertisement stored in the first space, wherein the service advertisement comprises information which is usable to access and execute a second service. The Examiner fails to provide any specific reasons for her rejection of claim 9. The Examiner only discusses combining Lehman with the ability to add functionality to a T Spaces server from Ford and also with XML. Appellants fail to see the relevance of these combinations to the specific limitations of claim 9.

Additionally, the Examiner has failed to cite any passage of either Ford or Lehman to support the rejection of claim 9. Instead, the Examiner states, without regard to any particular claim, “[t]o add an aspect of optional functionalities to the Lehman (‘420) T Spaces would have been obvious to one of ordinary skill in the art... as Tspace[s] provides a powerful mechanism for inter-process communication and synchronization” (Final Office Action, page 5, paragraph 9). The Examiner does not show what is meant by “optional functionalities” nor does claim 9 recite any “optional functionalities”. Thus, the proposed combination of Lehman in view of Ford and “optional functionalities” does not result in any system that teaches the limitations recited by claim 9.

Specifically, as noted above regarding the rejection of claim 1, Lehman in view of Ford does not teach or suggest the use of advertisements that include information usable to access and execute a service. While Appellants arguments above were regarding advertisements for spaces, those arguments also apply to advertisements for services in general. Instead, Lehman in view of Ford teaches the use of AddHandler() and NewTupleSpace() operators to add functionality to a T-Spaces server. However, nowhere does either Ford or Lehman mention reading a service advertisement stored in

the first space, wherein the service advertisement comprises information usable to access and execute a second service.

Furthermore, Lehman in view of Ford also fails to teach or suggest using the information in the service advertisement to execute the second service; generating a set of results of the second service in response to executing the second service, or publishing the set of results of the second service in the second space. Instead, Lehman teaches a Rhonda operator that allows to T-Spaces clients anonymously exchange information (Lehman, column 6, line 55 – column 7, line 24). However, Lehman in view of Ford does not disclose the Rhonda operator involving the use of information in a service advertisement to execute a second service. Nor does Lehman (or Ford) mention publishing a set of results of the second service in the second space. Thus, Lehman in view of Ford, whether combined with the Examiner's "optional functionalities" or not, clearly fails to teach or suggest the limitations of claim 9.

Claim 10:

As with the rejection of claim 9 discussed above, the Examiner fails to provide any specific reasons for her rejection of claim 10. The Examiner only discusses combining Lehman with the ability to add functionality to a T Spaces server from Ford and also with XML. Appellants fail to see the relevance of these combinations to the specific limitations of claim 10. Additionally, the Examiner states, without regard to any particular claim, "[t]o add an aspect of optional functionalities to the Lehman ('420) T Spaces would have been obvious to one of ordinary skill in the art... as Tspace[s] provides a powerful mechanism for inter-process communication and synchronization" (Final Office Action, page 5, paragraph 9). The Examiner does not show what is meant by "optional functionalities" nor does claim 10 recite any "optional functionalities". Thus, the proposed combination of Lehman in view of Ford and "optional functionalities" does not result in any system that teaches the limitations recited by claim 10.

In particular, regarding claim 10, Lehman in view of Ford clearly fails to teach or suggest that the creating the second space comprises creating a second address to the storage facility; and wherein accessing the second space comprises accessing the second space at the second address to the storage facility.

Instead, both Lehman and Ford teach that a command or request to a T Spaces server is a method invocation sent to the T Spaces server. The T Spaces server then determines an appropriate handler for the operator(s) in the request (Ford, column 7, lines 8-30, and Lehman, column 5, line 35- column 6, line 3). Additionally, Ford teaches that when adding a new operator, using the AddHandler() operator, to an existing T Spaces server, the new operator is added to an operator registry and a new handler for the new operator is installed in an appropriate handler factory (Ford, column 7, line 56-column 8, line 18). Lehman mentions the AddHandler() operator, but is silent regarding how it works.

Thus, Lehman and Ford both fail to teach or suggest that a new address is created when creating a new operator. Instead, all communication with a T Spaces server is directed to *a single address* so that an appropriate handler function from the Operator Registry may be determined. It would not make sense to create and use different addresses for new operators because the T Spaces server would then not be able to use the single Top-Level Command Handler to determine an appropriate handler for an operator. Having different addresses for different operators would prevent the Top-level Command Handler and the Operator Registry from performing correctly.

Claim 12, 17 and 22:

Regarding claim 12, contrary to the Examiner's contention, Lehman in view of Ford fails to teach or suggest accessing a first space, wherein information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space. Since

Lehman includes an almost identity discussion of T-Spaces as Ford, Lehman fails to overcome any of the deficiencies of Ford's teachings, as described above regarding the § 102(e) rejection of claim 12.

Specifically, Lehman in view of Ford fails to teach or suggest that information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space. Lehman, like Ford, teaches that T-Spaces clients include a Tuplespace class and a communications library to communicate with a T-Spaces server (Lehman, column 4, lines 55-61).

Also, Lehman clearly fails to teach or suggest a client requesting creation of a second space *by sending to the first space one of the messages specified by the first schema*. Lehman, like Ford, teaches the use of a specific operator, NewTupleSpace(), and that such commands originate as a method invocation on T-Spaces client (Lehman, column 5, lines 50-60). The examiner cites passages of Lehman that describe Lehman's Rhonda Operator that is used when two T Spaces clients exchange information via the T Spaces server (column 8, line 61-column 9, line 8, and column 9, lines 35-48) and that include broad statements regarding how "different computer programming languages, database systems, operating environments, and operating systems" could be substituted for those described by Lehman. Neither of the Examiner's cited passages has anything to do with creating a new space by sending a message specified by a schema.

Additionally, similarly to Ford, as described above regarding the § 102(e) rejection of claim 12, Lehman does not teach or suggest that the second space is initially configured to permit access only to the requesting client, nor wherein information usable to access the second space is provided in an advertisement for the second space, wherein the *advertisement for the second space comprises a second schema*, and wherein the second schema specifies one or more messages usable to invoke functions of the second space. As Lehman teaches essentially the identical T Spaces system as Ford, the

arguments above regarding the § 102(e) rejection of claim 12 as applied to Ford, apply to Lehman as well.

Claim 16:

Appellants note that the rejection of claim 16 is improper because claim 16 depends from claim 15 and the Examiner has not rejected claim 15 as being unpatentable over Lehman in view of Ford. As discussed below under the Third Ground of Rejection, the Examiner admits that Ford in view of Lehman fails to teach the limitations of claim 15 and relies upon a third prior art reference (Oliver) for the rejection of claim 15.

Further regarding claim 16, Ford in view of Lehman fails to disclose or suggest a second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service. The Examiner does not cite any particular passage of Lehman, but instead states, “[t]o add an aspect of optional functionalities to [Ford] would have been obvious ... as T Spaces provides a powerful mechanism for inter-process communication and synchronization.” However, Lehman’s extended functionality is described as providing a mechanism for exchanging information between two or more processes, providing synchronous, anonymous rendezvous and data exchange, providing a mechanism for controlling multiple processes, and providing barrier synchronization among a dynamically defined group of processes (Lehman, column 3, lines 21-30). Thus, while Lehman teaches that his Rhonda Operator can provide various means of inter-process communication, nowhere does Lehman describe anything regarding a second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service.

Claim 18:

Claim 18 was rejected “on the basis that Lehman teaches the use of different computer languages.” The Examiner’s cited reference in Lehman states, “different computer *programming* languages ... could be substituted for those described herein”

(Lehman, column 9, lines 10-14). Lehman is referring to programming languages that may be used to create programs, such as T-Spaces clients and servers, that may utilize his system and thus has no bearing on a *schema* that is expressed in a *data representation* language and that specifies messages usable to invoke functions of a space. The Examiner has responded to Appellants' previous arguments by pointing out the differences between a Data Manipulation Language and a Data Definition Language. Appellants submit however that the Examiner has misunderstood Appellants' argument. The issue is not whether a data representation language may be both static stored or dynamically created; instead, the issue is whether or not Lehman's general statement regarding *programming* languages suggests the use of a *data representation* language for a schema specifying messages usable to invoke functions of a space. Clearly, it does not.

Additionally, Appellants' arguments presented above regarding the rejection of claim 7 also apply to the rejection of claim 18.

Claim 19:

Regarding claim 19, the Examiner states that the motivation to combine XML with T-Spaces is noted within Lehman, "wherein the use of different computer languages is recognized as possible" and the Examiner further states that "as XML was in existence at the time of invention ... the use of the same as a programming language would have been obvious." As regarding claim 7, discussed above, the Examiner's cited reference states that "different computer *programming* languages ... could be substituted for those described" (Lehman, column 9, lines 10-14). Lehman is referring to *programming* languages that may be used to create programs that may utilize his invention and thus has no bearing on data representation languages, such as XML, which are used for very different purposes than computer programming languages. Furthermore, the Examiner's discussion regarding the traditional uses of XML is irrelevant. In the prior art, XML schemas have been used for describing data structures and content, not to specify messages usable to invoke functions of a space.

Additionally, the arguments presented above regarding claim 8 and specifically rebutting the Examiner's proposed combination of Lehman in view of Ford and XML apply to claim 19 as well.

Claim 20:

The Examiner fails to provide any specific reasons for her rejection of claim 20. The Examiner only discusses combining Lehman with the ability to add functionality to a T Spaces server from Ford and also with XML. Appellants fail to see the relevance of these combinations to the specific limitations of claim 20.

The Examiner has failed to cite any passage of either Ford or Lehman to support the rejection of claim 20. Instead, the Examiner states, without regard to any particular claim, "[t]o add an aspect of optional functionalities to the Lehman ('420) T Spaces would have been obvious to one of ordinary skill in the art... as Tspace[s] provides a powerful mechanism for inter-process communication and synchronization" (Final Office Action, page 5, paragraph 9). The Examiner does not show what is meant by "optional functionalities" nor does claim 20 recite any "optional functionalities". Thus, the proposed combination of Lehman in view of Ford and "optional functionalities" does not result in any system that includes the limitations recited by claim 20.

Specifically, as noted above regarding the rejection of claim 1, Lehman in view of Ford does not teach or suggest the use of advertisements that include information usable to access and execute a service. While Appellants arguments above were regarding advertisements for spaces, those arguments also apply to advertisements for services in general. Instead, Lehman in view of Ford teaches the use of an AddHandler() operator to add functionality to a T-Spaces server. However, nowhere does either Ford or Lehman mention reading a service advertisement stored in the first space, wherein the service advertisement comprises information usable to access and execute a second service.

Furthermore, Lehman in view of Ford fails to teach or suggest a client operable to read a service advertisement stored in the first space, wherein the service advertisement comprises information which is usable to access and execute the service; wherein the service is operable to generate a set of results of executing the service; create the second space; and publish the set of results in the second space. Ford and Lehman, whether separately or in combination, do not teach such functionality.

Thus, Lehman in view of Ford, whether combined with the Examiner's "optional functionalities" or not, clearly fails to teach or suggest all the limitations of claim 20.

Claim 21:

Regarding claim 21, Ford in view of Lehman fails to teach or suggest wherein in requesting creation of the second space, the first client is operable to request creation of a second address to the storage facility; and wherein in accessing the second space, the first client is operable to access the second space at the second address to the storage facility. Instead, both Ford and Lehman teach that a command or request to a T Spaces server is a method invocation sent to the T Spaces server. The T Spaces server then determines an appropriate handler for the operator(s) in the request (Ford, column 7, lines 8-30, and Lehman, column 5, line 35- column 6, line 3).

Additionally, Ford teaches that when adding a new operator, using the AddHandler() operator, to an existing T Spaces server, the new operator is added to an operator registry and a new handler for the new operator is installed in an appropriate handler factory (Ford, column 7, line 56-column 8, line 18). Lehman mentions the AddHandler() operator, but is silent regarding how it works. Thus, Lehman in view of Ford does not teach that a new address is created when creating a new operator. Instead, all communication with a T Spaces server is directed to *a single address* so that an appropriate handler function from the Operator Registry may be determined. It would not make sense to create and use different addresses for new operators because the T Spaces server would then not be able to use the single Top-Level Command Handler to

determine an appropriate handler for an operator. Having different addresses for different operators would prevent the Top-level Command Handler and the Operator Registry from performing correctly.

Furthermore, as with the rejection of claim 10 discussed above, the Examiner fails to provide any specific reasons for her rejection of claim 21. The Examiner only discusses combining Lehman with the ability to add functionality to a T Spaces server from Ford and also with XML. Appellants fail to see the relevance of these combinations to the specific limitations of claim 21. Additionally, the Examiner states, without regard to any particular claim, “[t]o add an aspect of optional functionalities to the Lehman (‘420) T Spaces would have been obvious to one of ordinary skill in the art... as Tspace[s] provides a powerful mechanism for inter-process communication and synchronization” (Final Office Action, page 5, paragraph 9). The Examiner does not show what is meant by “optional functionalities” nor does claim 21 recite any “optional functionalities”. Thus, the proposed combination of Lehman in view of Ford and “optional functionalities” does not result in a system that teaches all the limitations recited by claim 21.

Claim 23, 28 and 33:

Regarding claim 23, contrary to the Examiner’s contention, Lehman in view of Ford fails to teach or suggest accessing a first space, wherein information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space. Since Lehman includes an almost identical discussion of T-Spaces as Ford, Lehman fails to overcome any of the deficiencies of Ford’s teachings, as described above regarding the § 102(e) rejection of claim 23.

Specifically, Lehman in view of ford fails to teach or suggest that information usable to access the first space is provided in an advertisement for the first space, wherein

the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space. Lehman, like Ford, teaches that T-Spaces clients include a Tuplespace class and a communications library to communicate with a T-Spaces server (Lehman, column 4, lines 55-61).

Also, Lehman clearly fails to teach or suggest a client requesting creation of a second space *by sending to the first space one of the messages specified by the first schema*. Lehman, like Ford, teaches the use of a specific operator, NewTupleSpace(), and that such commands originate as a method invocation on T-Spaces client (Lehman, column 5, lines 50-60). The examiner cites passages of Lehman that describe Lehman's Rhonda Operator that is used when two T Spaces clients exchange information via the T Spaces server (column 8, line 61-column 9, line 8, and column 9, lines 35-48) and that include broad statements regarding how "different computer programming languages, database systems, operating environments, and operating systems" could be substituted for those described by Lehman. Neither of the Examiner's cited passages refers to creating a new space by sending a message specified by a schema.

Additionally, similarly to Ford, as described above regarding the § 102(e) rejection of claim 23, Lehman does not teach or suggest that the second space is initially configured to permit access only to the requesting client, nor wherein information usable to access the second space is provided in an advertisement for the second space, wherein the *advertisement for the second space comprises a second schema*, and wherein the second schema specifies one or more messages usable to invoke functions of the second space. As Lehman teaches essentially the identical T Spaces system as Ford, the arguments above regarding the § 102(e) rejection of claim 23 as applied to Ford, apply to Lehman as well.

Claim 27:

Appellants note that the rejection of claim 27 is improper because claim 27 depends from claim 26 and the Examiner has not rejected claim 26 as being unpatentable over Ford in view of Lehman. As discussed below under the Third Ground of Rejection, the Examiner admits that Ford in view of Lehman fails to teach the limitations of claim 26 and relies upon Oliver for the rejection of claim 26.

Further regarding claim 27, Ford in view of Lehman fails to disclose or suggest that the program instructions are further computer-executable to implement the second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service. The Examiner does not cite any particular passage of Lehman, but instead states, “[t]o add an aspect of optional functionalities to [Ford] would have been obvious ... as T Spaces provides a powerful mechanism for inter-process communication and synchronization.” However, Lehman’s extended functionality is described as providing a mechanism for exchanging information between two or more processes, providing synchronous, anonymous rendezvous and data exchange, providing a mechanism for controlling multiple processes, and providing barrier synchronization among a dynamically defined group of processes (Lehman, column 3, lines 21-30). Thus, while Lehman teaches that his Rhonda Operator can provide various means of inter-process communication, nowhere does Lehman describe anything regarding a second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service.

Claim 29:

Claim 29 was rejected “on the basis that Lehman teaches the use of different computer languages.” The Examiner’s cited reference in Lehman states, “different computer *programming* languages ... could be substituted for those described herein” (Lehman, column 9, lines 10-14). Appellants assert that Lehman is referring to programming languages that may be used to create programs, such as T-Spaces clients and servers, that may utilize his system and thus has no bearing on a *schema* that is expressed

in a *data representation* language and that specifies messages usable to invoke functions of a space. The Examiner has responded to Appellants' previous arguments by pointing out the differences between a Data Manipulation Language and a Data Definition Language. Appellants submit however that the Examiner has misunderstood Appellants' argument. The issue is not whether a data representation language may be both static stored or dynamically created; instead, the issue is whether or not Lehman's general statement regarding *programming* languages suggests the use of a *data representation* language for a schema specifying messages usable to invoke functions of a space. Clearly, it does not.

Additionally, the arguments presented above regarding the rejection of claim 7 also apply to the rejection of claim 29.

Claim 30:

Regarding claim 30, the Examiner states that the motivation to combine XML with T-Spaces is noted within Lehman, "wherein the use of different computer languages is recognized as possible" and the Examiner further states that "as XML was in existence at the time of invention ... the use of the same as a programming language would have been obvious." As regarding claim 7, discussed above, the Examiner's cited reference states that "different computer *programming* languages ... could be substituted for those described" (Lehman, column 9, lines 10-14). Lehman is referring to *programming* languages that may be used to create programs that may utilize his invention and thus has no bearing on data representation languages, such as XML, which are used for very different purposes than computer programming languages. Furthermore, the Examiner's discussion regarding the traditional uses of XML is irrelevant. In the prior art, XML schema have been used to describe data structures, not to specify messages usable to invoke functions of a space.

Furthermore, Appellants' arguments presented above regarding the rejection of claim 8 also apply to the rejection of claim 29.

Claim 31:

Regarding claim 31, Lehman in view of Ford does not teach or suggest that the program instructions are further computer-executable to implement: reading a service advertisement stored in the first space, wherein the service advertisement comprises information which is usable to access and execute a second service. The Examiner fails to provide any specific reasons for her rejection of claim 31. The Examiner only discusses combining Lehman with the ability to add functionality to a T Spaces server from Ford and also with XML. Appellants fail to see the relevance of these combinations to the specific limitations of claim 31.

Additionally, the Examiner has failed to cite any passage of either Ford or Lehman to support the rejection of claim 9. Instead, the Examiner states, without regard to any particular claim, “[t]o add an aspect of optional functionalities to the Lehman (‘420) T Spaces would have been obvious to one of ordinary skill in the art... as Tspace[s] provides a powerful mechanism for inter-process communication and synchronization” (Final Office Action, page 5, paragraph 9). The Examiner does not show what is meant by “optional functionalities” nor does claim 31 recite any “optional functionalities”. Thus, the proposed combination of Lehman in view of Ford and “optional functionalities” does not result in a system that teaches all the limitations recited by claim 31.

Additionally, since claim 31 recites a carrier medium comprising program instructions that are computer-executable to implement the method similar to that recited in claim 9, Appellants arguments above regarding the rejection of claim 9 also apply to the rejection of claim 31.

Claim 32:

As with the rejection of claim 31 discussed above, the Examiner fails to provide any specific reasons for her rejection of claim 32. The Examiner only discusses

combining Lehman with the ability to add functionality to a T Spaces server from Ford and also with XML. Appellants fail to see the relevance of these combinations to the specific limitations of claim 32. Additionally, the Examiner states, without regard to any particular claim, “[t]o add an aspect of optional functionalities to the Lehman (‘420) T Spaces would have been obvious to one of ordinary skill in the art... as Tspace[s] provides a powerful mechanism for inter-process communication and synchronization” (Final Office Action, page 5, paragraph 9). The Examiner does not show what is meant by “optional functionalities” nor does claim 32 recite any “optional functionalities”. Thus, the proposed combination of Lehman in view of Ford and “optional functionalities” does not result in a system that teaches all the limitations recited by claim 32.

Lehman, in view of Ford clearly fails to teach or suggest that the creating the second space comprises creating a second address to the storage facility; and wherein accessing the second space comprises accessing the second space at the second address to the storage facility. Instead, both Lehman and Ford teach that a command or request to a T Spaces server is a method invocation sent to the T Spaces server. The T Spaces server then determines an appropriate handler for the operator(s) in the request (Ford, column 7, lines 8-30, and Lehman, column 5, line 35- column 6, line 3). Additionally, Ford teaches that when adding a new operator, using the AddHandler() operator, to an existing T Spaces server, the new operator is added to an operator registry and a new handler for the new operator is installed in an appropriate handler factory (Ford, column 7, line 56- column 8, line 18). Lehman mentions the AddHandler() operator, but is silent regarding how it works.

Thus, Lehman and Ford both fail to teach or suggest that a new address is created to create a new operator. Instead, all communication with a T Spaces server is directed to *a single address* so that an appropriate handler function from the Operator Registry may be determined. It would not make sense to create and use different addresses for new operators because the T Spaces server would then not be able to use the single Top-Level Command Handler to determine an appropriate handler for an operator. Having different

addresses for different operators would prevent the Top-level Command Handler and the Operator Registry from performing correctly.

Third Ground of Rejection:

Claims 2-4, 13-15 and 24-26 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Lehman in view of Oliver (U.S. Publication 2002/0133412). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claim 2:

Claim 2 is allowable for at least the reasons presented herein regarding its independent base claim, claim 1.

Claim 3:

Regarding claim 3, the Examiner presumably contends that Lehman in view of Ford in further view of Oliver teaches a client sending the root authentication token to a second client; the second client accessing the second space by sending to the second space one of the messages specified by the second schema. Appellants, however, disagree with the Examiner's contention. Lehman, Ford and Oliver clearly fail to teach or suggest sending the root authentication token to a second client and further fail to teach the second client accessing the second space by sending to the second space one of the messages specified by the second schema.

The Examiner admits that Lehman does not teach the use of root authentication tokens. Ford also fails to teach the use of root authentication tokens. Oliver teaches a system for managing client accounts and controlling access to resources (Oliver, abstract) that uses authentication tokens; however, Oliver fails to teach one client sending an authentication token to a second client. In contrast, Oliver teaches that an authentication token contains information that identifies the associated user (Oliver, paragraph 126).

Hence, it would not make sense for a client to send such an authentication token to a second client, as the authentication token would not then properly identify the second client.

The Examiner has responded in the Response to Arguments section that Ford and Lehman teach the use of keys in the form of IDs and that when combined with the authentication tokens of Oliver results in a system wherein the tokens of Oliver are used to transfer the keys of Ford and Lehman. This is clearly a hindsight-based application of the art. The only keys taught in either Ford or Lehman refer to request identifiers used as keys to route responses from a T-Spaces server back to the originating T-Client application thread (Lehman, column 5, lines 13-23 and Ford, column 6, lines 51-52). Additionally, the only other identifiers taught either by Ford or Lehman are T-Space client identifiers (Ford, column 7, lines 31-37 and Lehman, column 5, lines 60-65). Neither the request identifiers or T-Client identifiers can be transferred from one client to another as they are meaningless with respect to the other client and would only confuse and corrupt the T-Spaces server system or cause responses from the server to be misrouted. Thus, the Examiner's proposed combination of Lehman, Ford and Oliver would not result in a system that comprises the requesting client sending the root authentication token to a second client. Instead, such a combination would only result in a T Spaces system that used the authentication tokens of Oliver to indicate the access control privileges of Ford and Lehman.

Further, as shown above, neither Lehman nor Ford, separately or in combination, teach or suggest the use of a schema specifying messages usable to invoke functions of a space, nor do they teach accessing a space by sending messages specified in such a schema. Oliver teaches the use of authentication tokens, but also fails to teach anything regarding a schema specifying messages usable to invoke functions of a space. Thus, the combination of Lehman, Ford and Oliver does not teach or suggest a client sending the authentication token to a second client and the second client accessing the second space by sending to the second space one of the messages specified by the second schema.

Claim 4:

Regarding claim 4, Lehman, Ford and Oliver, individually and in any combination, do not teach or suggest a client modifying a security policy of the second space, whereby the second space is configured to permit access to a second client, as the Examiner implies. As the Examiner has failed to cite any particular portions of Lehman, Ford or Oliver to support the broad statement that claim 4 is rejected in light of the teachings and motivations of claims 1 and 2, as they “refer to the use and manipulation of security measures including, but not limited to, authentication means” (Office Action, page 7, lines 9-12). However, any authentication means taught by Lehman, Ford and Oliver fails to include a requesting client modifying a security policy of a space, whereby the second space is configured to permit access to a second client. Ford and Lehman only refer to the fact that “users can establish security policies by setting user and group permissions on a Tuplespace basis” (Ford, column 5, lines 10-12). Oliver includes a system for managing client accounts and controlling access to resources over data networks wherein a client registered with one service provider is allowed to access the resources of other service providers of the system (Oliver, paragraph 17). Nowhere, however, does Oliver describe a requesting client modifying a security policy for a space, whereby the second space is configured to permit access to a second client.

Claim 13:

Claim 13 is allowable for at least the reasons presented herein regarding its independent base claim, claim 12.

Claim 14:

Regarding claim 14, the Examiner presumably contends that Lehman in view of Ford in further view of Oliver teaches a client sending the root authentication token to a second client; the second client accessing the second space by sending to the second space one of the messages specified by the second schema. Appellants, however, disagree with the Examiner’s contention. Lehman, Ford and Oliver clearly fail to teach

or suggest sending the root authentication token to a second client and further fail to teach or suggest the second client accessing the second space by sending to the second space one of the messages specified by the second schema.

The Examiner admits that Lehman does not teach the use of root authentication tokens. Ford also fails to teach the use of root authentication tokens. Oliver teaches a system for managing client accounts and controlling access to resources (Oliver, abstract) that uses authentication tokens; however, Oliver fails to teach or suggest one client sending an authentication token to a second client. In contrast, Oliver teaches that an authentication token contains information that identifies the associated user (Oliver, paragraph 126). Hence, it would not make sense for a client to send such an authentication token to a second client, as the authentication token would not then properly identify the second client.

The Examiner has responded in the Response to Arguments section that Ford and Lehman teach the use of keys in the form of IDs and that when combined with the authentication tokens of Oliver results in a system wherein the tokens of Oliver are used to transfer the keys of Ford and Lehman. This is clearly a hindsight-based application of the art. The only keys taught in either Ford or Lehman refer to request identifiers used as keys to route responses from a T-Spaces server back to the originating T-Client application thread (Lehman, column 5, lines 13-23 and Ford, column 6, lines 51-52). Additionally, the only other identifiers taught either by Ford or Lehman are T-Space client identifiers (Ford, column 7, lines 31-37 and Lehman, column 5, lines 60-65). Neither the request identifiers or T-Client identifiers can be transferred from one client to another as they are meaningless with respect to the other client and would only confuse and corrupt the T-Spaces server system or cause responses from the server to be misrouted. Thus, the Examiner's proposed combination of Lehman, Ford, and Oliver would not result in a system that comprises the requesting client sending the root authentication token to a second client. Instead, such a combination would only result in a T Spaces system that used the authentication tokens of Oliver to indicate the access control privileges of Ford and Lehman.

Further, as shown above, neither Lehman nor Ford, separately or in combination, teach or suggest the use of a schema specifying messages usable to invoke functions of a space, nor do they teach accessing a space by sending messages specified in such a schema. Oliver teaches the use of authentication tokens, but also fails to teach anything regarding a schema specifying messages usable to invoke functions of a space. Thus, the combination of Lehman, Ford and Oliver does not teach or suggest a client sending the authentication token to a second client and the second client accessing the second space by sending to the second space one of the messages specified by the second schema.

Claim 15:

Regarding claim 15, Lehman, Ford and Oliver, individually and in any combination, do not teach or suggest a client modifying a security policy of the second space, whereby the second space is configured to permit access to a second client, as the Examiner implies. As the Examiner has failed to cite any particular portions of Lehman, Ford or Oliver to support the broad statement that claim 4 is rejected in light of the teachings and motivations of claims 1 and 2, as they “refer to the use and manipulation of security measures including, but not limited to, authentication means” (Office Action, page 7, lines 9-12). However, any authentication means taught by Lehman, Ford and Oliver fails to include a requesting client modifying a security policy of a space, whereby the second space is configured to permit access to a second client. Ford and Lehman only refer to the fact that “users can establish security policies by setting user and group permissions on a Tuplespace basis” (Ford, column 5, lines 10-12). Oliver includes a system for managing client accounts and controlling access to resources over data networks wherein a client registered with one service provider is allowed to access the resources of other service providers of the system (Oliver, paragraph 17). Nowhere, however, does Oliver describe or suggest a requesting client modifying a security policy for a space, whereby the second space is configured to permit access to a second client.

Claim 24:

Claim 24 is allowable for at least the reasons presented herein regarding its independent base claim, claim 23.

Claim 25:

Regarding claim 25, the Examiner presumably contends that Lehman in view of Ford in further view of Oliver teaches a client sending the root authentication token to a second client; the second client accessing the second space by sending to the second space one of the messages specified by the second schema. Appellants, however, disagree with the Examiner's contention. Lehman, Ford and Oliver clearly fail to teach or suggest sending the root authentication token to a second client and further fail to teach the second client accessing the second space by sending to the second space one of the messages specified by the second schema.

The Examiner admits that Lehman does not teach the use of root authentication tokens. Ford also fails to teach the use of root authentication tokens. Oliver teaches a system for managing client accounts and controlling access to resources (Oliver, abstract) that uses authentication tokens; however, Oliver fails to teach one client sending an authentication token to a second client. In contrast, Oliver teaches that an authentication token contains information that identifies the associated user (Oliver, paragraph 126). Hence, it would not make sense for a client to send such an authentication token to a second client, as the authentication token would not then properly identify the second client.

The Examiner has responded in the Response to Arguments section that Ford and Lehman teach the use of keys in the form of IDs and that when combined with the authentication tokens of Oliver results in a system wherein the tokens of Oliver are used to transfer the keys of Ford and Lehman. This is clearly a hindsight-based application of the art. The only keys taught in either Ford or Lehman refer to request identifiers used as keys to route responses from a T-Spaces server back to the originating T-Client application thread (Lehman, column 5, lines 13-23 and Ford, column 6, lines 51-52).

Additionally, the only other identifiers taught either by Ford or Lehman are T-Space client identifiers (Ford, column 7, lines 31-37 and Lehman, column 5, lines 60-65). Neither the request identifiers or T-Client identifiers can be transferred from one client to another as they are meaningless with respect to the other client and would only confuse and corrupt the T-Spaces server system or cause responses from the server to be misrouted. Thus, the Examiner's proposed combination of Lehman, Ford, and Oliver would not result in a system that comprises the requesting client sending the root authentication token to a second client. Instead, such a combination would only result in a T Spaces system that used the authentication tokens of Oliver to indicate the access control privileges of Ford and Lehman.

Further, as shown above, neither Lehman nor Ford, separately or in combination, teach or suggest the use of a schema specifying messages usable to invoke functions of a space, nor do they teach accessing a space by sending messages specified in such a schema. Oliver teaches the use of authentication tokens, but also fails to teach anything regarding a schema specifying messages usable to invoke functions of a space. Thus, the combination of Lehman, Ford and Oliver does not teach or suggest a client sending the authentication token to a second client and the second client accessing the second space by sending to the second space one of the messages specified by the second schema.

Claim 26:

Regarding claim 26, Lehman, Ford and Oliver, individually and in any combination, do not teach or suggest a client modifying a security policy of the second space, whereby the second space is configured to permit access to a second client, as the Examiner implies. The Examiner has failed to cite any particular portions of Lehman, Ford or Oliver to support the broad statement that claim 4 is rejected in light of the teachings and motivations of claims 1 and 2, as they "refer to the use and manipulation of security measures including, but not limited to, authentication means" (Office Action, page 7, lines 9-12). However, any authentication means taught by Lehman, Ford and Oliver fails to include a requesting client modifying a security policy of a space, whereby

the second space is configured to permit access to a second client. Ford and Lehman only refer to the fact that “users can establish security policies by setting user and group permissions on a Tuplespace basis” (Ford, column 5, lines 10-12). Oliver includes a system for managing client accounts and controlling access to resources over data networks wherein a client registered with one service provider is allowed to access the resources of other service providers of the system (Oliver, paragraph 17). Nowhere, however, does Oliver describe or suggest a requesting client modifying a security policy for a space, whereby the second space is configured to permit access to a second client.

Fourth Ground of Rejection:

Claims 1-33 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Lehman in view of Ford and in further view of “IBM Systems Journal, Vol. 37, No. 3” by Wyckoff, McLaughry, Lehman and Ford, 1998 (hereinafter “Wyckoff”). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Regarding the rejection of claims 1-33 over Lehman in view of Ford in further view of Wyckoff, the Examiner fails to include any specific arguments for the rejection any of claims 1-33. Therefore, the Examiner has clearly failed to state a *prima facie* rejection. The Examiner asserts only that Lehman and Ford “disclose every claim limitation with the exception of the incorporation of XML” (Final Office Action, page 8, paragraph 18). However, the Examiner has already admitted that Lehman in view of Ford fail to teach the limitations of claims 2-4, 13-15, and 24-26 (*See e.g.* Final Office Action, page 6, paragraph 13 and the above discussion regarding the Second Ground of Rejection). Furthermore, Wyckoff gives an introduction to both Tuplespaces and T Spaces that is largely identical to both Lehman and Ford. Wyckoff thus fails to teach anything that overcomes the deficiencies of Lehman and Ford, as described herein above.

Claim 1, 6 and 11:

Regarding claim 1, Appellants' arguments given above regarding Lehman in view of Ford regarding claim 1 apply here as well. In addition, Wyckoff presents an introduction to both Tuplespaces and T Spaces that is largely identical to both Lehman and Ford. Wyckoff fails to teach anything that overcomes the deficiencies of Lehman and Ford, as described herein above. The Examiner relies upon Wyckoff to teach access controls that include security policies. However, Wyckoff fails to disclose creating the second space in response to the requesting client requesting creation of the second space, wherein the second space is initially configured to permit access only to the requesting client. The Examiner cites a passage of Wyckoff that only refers to users being able to establish security policies by setting user and group permissions on a Tuplespace basis (Wyckoff, page 7, access controls). However, the fact that user can set permissions when establishing security policies does not suggest that when created a second space is initially configured to permit access only to the requesting client.

Additionally, Appellants' arguments as presented above regarding the rejection of claim 1 over Lehman in view of Ford in the Second Grounds of Rejection apply to the rejection of claim 1 over Lehman in view of Ford with equal force.

Claim 2:

Regarding claim 2, contrary to the Examiner's assertion, Lehman in view of Ford in further view of Wyckoff does not teach or suggest creating a root authentication token for the second space; initializing an authentication service associated with the second space, wherein the second space is configured to permit access only to a client holding the root authentication token, and sending the root authentication token to the requesting client. The Examiner has admitted that Lehman and Ford fail to teach the user of root authentication tokens (Office Action, page 7, lines 4-5). The Examiner argues that Wyckoff's discussion of the access controls of T-Spaces "would include specifically enumerated by Applicant." However, the portion of Wyckoff cited by the Examiner (Wyckoff, page 7, "T Spaces overview") only states, "[u]sers [of T-Spaces] can establish security policies by setting user and group permissions on a Tuplespace basis." Setting

user and group permissions are very different than creating and using a root authentication token. Wyckoff also fails to mention anything regarding authentication tokens. Instead Wyckoff only refers to a general ability for user and group-based access control without teaching anything about authentication tokens.

Claim 3:

Regarding claim 3, Lehman in view of Ford in further view of Wyckoff fails to teach or suggest the requesting client sending the root authentication token to the second client and the second client access the second space by sending to the second space one of the messages specified by the second schema. Ford and Lehman, both singly and in combination, fail to disclose or suggest sending an authentication token to a second client, as shown above, and Wyckoff does not teach or suggest anything at all regarding authentication tokens and thus clearly does not disclose one client sending an authentication token to a second client. A client sending an authentication token to another client does not make sense in the T Spaces system, as outlined above. Furthermore, Wyckoff also fails to disclose the second client access the second space by sending to the second space one of the messages specified by the second schema. As with Ford and Lehman, Wyckoff does not mention anything about a schema specifying messages usable to invoke functions of a space, and also fails to mention a second client sending such a message to a second space.

Claim 4:

Regarding claim 4, Lehman, Ford and Wyckoff, individually and in any combination, do not teach or suggest a client modifying a security policy of the second space, whereby the second space is configured to permit access to a second client, as the Examiner implies. The Examiner has failed to cite any particular portion in Wyckoff, Lehman, or Ford, but only states that Wyckoff teaches access controls that include security policies. However, any access controls taught by Lehman, Ford and Wyckoff fail to include a requesting client modifying a security policy of a space, whereby the second space is configured to permit access to a second client. The T Spaces system only

allows users to establish security policies by setting user and group permissions on a Tuplespace basis (e.g. Ford, column 5, lines 10-12). Wyckoff does not include any additional access control features that involve a requesting client modifying a security policy for a space, whereby the second space is configured to permit access to a second client.

Claim 5:

Further regarding claim 5, Ford in view of Lehman in further view of Wyckoff fails to disclose or suggest a second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service. The Examiner does not cite any particular passage of Ford, Lehman, or Wyckoff that teaches a second client reading a service advertisement from the second space. As Wyckoff fails to overcome any of the deficiencies of Ford in view of Lehman, the arguments and remarks herein above regarding the rejections of claim 5 over Lehman in view of Ford (Second Ground of Rejection) apply here as well.

Claim 7:

Regarding claim 7, Ford in view of Lehman in further view of Wyckoff fails to teach or suggest wherein the first schema is expressed in a data representation language; and wherein the second schema is expressed in the data representation language. Wyckoff does not provide any teachings regarding the use of a data representation language for expressing message schemas and thus does not overcome any of the deficiencies of Ford and Lehman. Thus, Appellants' arguments herein above regarding the rejection of claim 7 over Lehman in view of Ford (Second Ground of Rejection) also apply here with equal force.

Claim 8:

Regarding claim 8, Ford in view of Lehman in further view of Wyckoff fails to teach or suggest that the data representation language comprises eXtensible Markup

Language (XML). As shown above regarding the rejection of claim 8 over Lehman in view of Ford (Second Ground of Rejection), neither Ford nor Lehman teach anything regarding the use of XML. Wyckoff fails to disclose anything regarding XML and the Examiner does not rely upon Wyckoff to teach XML. The Examiner maintains that it would be obvious to include XML into the teachings of Ford and Lehman. However, as discussed above (rejection of claim 8, Second Ground of Rejection) not only is it non-obvious to include the use of XML for expressing message schemas, the Examiner has not provided a proper motivation for including XML in the teachings of Ford and Lehman for expressing message schemas. Please refer to Appellants' arguments above regarding the rejection of claim 8 in the Second Ground of Rejection as they apply here as well.

Claim 9:

Regarding claim 9, Ford in view of Lehman in further view of Wyckoff does not teach or suggest reading a service advertisement stored in the first space, wherein the service advertisement comprises information which is usable to access and execute a second service, as the Examiner contends. Furthermore, Lehman in view of Ford and Wyckoff also fails to teach or suggest using the information in the service advertisement to execute the second service; generating a set of results of the second service in response to executing the second service, or publishing the set of results of the second service in the second space. Please refer to Appellants' arguments above regarding the rejection of claim 9 in the Second Ground of Rejection as they apply here as well. Wyckoff fails to overcome any of the deficiencies outlined above regarding the rejection of claim 9.

Claim 10:

Regarding claim 10, Ford in view of Lehman in further view of Wyckoff clearly fails to teach or suggest wherein creating the second space comprises creating a second address to the storage facility; and wherein accessing the second space comprises accessing the second space at the second address to the storage facility. Instead, Ford, Lehman and Wyckoff teach that a command or request to a T Spaces server is a method

invocation sent to the T Spaces server. The T Spaces server then determines an appropriate handler for the operator(s) in the request (Ford, column 7, lines 8-30, and Lehman, column 5, line 35- column 6, line 3).

Additionally, Ford teaches that when adding a new operator, using the AddHandler() operator, to an existing T Spaces server, the new operator is added to an operator registry and a new handler for the new operator is installed in an appropriate handler factory (Ford, column 7, line 56-column 8, line 18). Lehman and Wyckoff mention the AddHandler() operator, but are silent regarding how it works. Thus, Ford, Lehman and Wyckoff do not teach that a new address is created to create a new operator. Instead, all communication with a T Spaces server is directed to *a single address* so that an appropriate handler function from the Operator Registry may be determined. It would not make sense to create and use different addresses for new operators because the T Spaces server would then not be able to use the single Top-Level Command Handler to determine an appropriate handler for an operator. Having different addresses for different operators would prevent the Top-level Command Handler and the Operator Registry from performing correctly.

Claim 12, 17 and 22:

Regarding claim 12, Appellants' arguments given above regarding Lehman in view of Ford regarding claim 12 apply here as well. In addition, Wyckoff presents an introduction to both Tuplespaces and T Spaces that is largely identical to both Lehman and Ford. Wyckoff fails to teach anything that overcomes the deficiencies of Lehman and Ford, as described herein above. The Examiner relies upon Wyckoff to teach access controls that include security policies. However, Wyckoff fails to disclose or suggest creating the second space in response to the requesting client requesting creation of the second space, wherein the second space is initially configured to permit access only to the requesting client. The Examiner cites a passage of Wyckoff that only refers to users being able to establish security policies by setting user and group permissions on a Tuplespace basis (Wyckoff, page 7, access controls). However, the fact that user can set

permissions when establishing security policies does not suggest that when created a second space is initially configured to permit access only to the requesting client. Furthermore the arguments above regarding the rejection of claim 1 over Ford in view of Lehman in further view of Wyckoff also apply here with equal force.

Claim 13:

Regarding claim 13, contrary to the Examiner's assertion, Lehman in view of Ford in further view of Wyckoff does not teach or suggest creating a root authentication token for the second space; initializing an authentication service associated with the second space, wherein the second space is configured to permit access only to a client holding the root authentication token, and sending the root authentication token to the requesting client. The Examiner has admitted that Lehman and Ford fail to teach the user of root authentication tokens (Office Action, page 7, lines 4-5). Wyckoff also fails to mention anything regarding authentication tokens. Instead Wyckoff only refers to a general ability for user and group-based access control without teaching anything about authentication tokens. Please refer to the remarks above regarding the rejection of claim 2 as they apply to claim 13 as well.

Claim 14:

Regarding claim 14, Lehman in view of Ford in further view of Wyckoff fails to teach or suggest the requesting client sending the root authentication token to the second client and the second client access the second space by sending to the second space one of the messages specified by the second schema. Ford and Lehman, both singly and in combination, fail to disclose sending an authentication token to a second client, as shown above, and Wyckoff does not teach anything at all regarding authentication token and clearly does not disclose one client sending an authentication token to a second client. A client sending an authentication token to another client does not make sense in the T Spaces system, as outlined above. Furthermore, Wyckoff also fails to disclose the second client access the second space by sending to the second space one of the messages specified by the second schema. As with Ford and Lehman, Wyckoff does not mention

anything about a schema specifying messages usable to invoke functions of a space, and also fails to mention a second client sending such a message to a second space. Please refer to the remarks above regarding the rejection of claim 3 as they apply to claim 14 as well.

Claim 15

Regarding claim 15, Lehman, Ford and Wyckoff, individually and in any combination, do not teach or suggest a client modifying a security policy of the second space, whereby the second space is configured to permit access to a second client, as the Examiner implies. The Examiner has failed to cite any particular portion in Wyckoff Lehman, or Ford, but only states that Wyckoff teaches access controls that include security policies. However, any access controls taught by Lehman, Ford and Wyckoff fail to include a requesting client modifying a security policy of a space, whereby the second space is configured to permit access to a second client. The T Spaces system only allows users to establish security policies by setting user and group permissions on a Tuplespace basis (e.g. Ford, column 5, lines 10-12). Wyckoff does not include any additional access control features that involve a requesting client modifying a security policy for a space, whereby the second space is configured to permit access to a second client. Please note that Appellants' arguments above regarding claim 4 also apply to claim 15.

Claim 16:

Regarding claim 16, Ford in view of Lehman in further view of Wyckoff fails to disclose or suggest a second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service. Please note that Appellants' arguments above regarding claim 5 also apply to claim 16.

Claim 18:

In regard to claim 18, Ford in view of Lehman in further view of Wyckoff fails to teach or suggest that the first schema is expressed in a data representation language; and wherein the second schema is expressed in the data representation language. Please refer to Appellants' arguments above regarding claim 7 as they also apply to claim 18.

Claim 19:

Regarding claim 19, Ford in view of Lehman in further view of Wyckoff fails to teach or suggest that the data representation language comprises eXtensible Markup Language (XML). Please refer to Appellants' arguments above regarding claim 8 as they also apply to claim 19.

Claim 20:

Regarding claim 20, Ford in view of Lehman in further view of Wyckoff does not teach or suggest reading a service advertisement stored in the first space, wherein the service advertisement comprises information which is usable to access and execute a second service, as the Examiner contends. Furthermore, Lehman in view of Ford and Wyckoff also fails to teach using the information in the service advertisement to execute the second service; generating a set of results of the second service in response to executing the second service, or publishing the set of results of the second service in the second space. Please refer to Appellants' arguments above regarding claim 9 as they apply here with equal force.

Claim 21:

Regarding claim 21, Ford in view of Lehman in further view of Wyckoff does not teach or suggest wherein in accessing the first space, the first client is operable to access the first space at a first address to a storage facility; wherein in requesting creation of the second space, the first client is operable to request creation of a second address to the storage facility; and wherein in accessing the second space, the first client is operable to access the second space at the second address to the storage facility. Instead, Ford,

Lehman and Wyckoff teach that a command or request to a T Spaces server is a method invocation sent to the T Spaces server. The T Spaces server then determines an appropriate handler for the operator(s) in the request (Ford, column 7, lines 8-30, and Lehman, column 5, line 35- column 6, line 3). Please note that Appellants' arguments above regarding claim 10 apply to claim 21 with equal force.

Claim 23, 28 and 33:

Regarding claim 23, Ford in view of Lehman in further view of Wyckoff fails to disclose or suggest creating the second space in response to the requesting client requesting creation of the second space, wherein the second space is initially configured to permit access only to the requesting client. The Examiner cites a passage of Wyckoff that only refers to users being able to establish security policies by setting user and group permissions on a Tuplespace basis (Wyckoff, page 7, access controls). However, the fact that user can set permissions when establishing security policies does not suggest that when created a second space is initially configured to permit access only to the requesting client. Furthermore the arguments above regarding the rejection of claim 1 over Ford in view of Lehman in further view of Wyckoff also apply here with equal force.

Please note that Appellants' arguments given above regarding claims 1 and 12 apply here as well.

Claim 24:

Regarding claim 24, contrary to the Examiner's assertion, Lehman in view of Ford in further view of Wyckoff does not teach or suggest creating a root authentication token for the second space; initializing an authentication service associated with the second space, wherein the second space is configured to permit access only to a client holding the root authentication token, and sending the root authentication token to the requesting client. The Examiner has admitted that Lehman and Ford fail to teach the user of root authentication tokens (Office Action, page 7, lines 4-5). Wyckoff also fails to mention anything regarding authentication tokens. Instead Wyckoff only refers to a

general ability for user and group-based access control without teaching anything about authentication tokens. Please refer to the remarks above regarding the rejection of claims 2 and 13 as they apply to claim 24 as well.

Claim 25:

Regarding claim 25, Lehman in view of Ford in further view of Wyckoff fails to teach or suggest the requesting client sending the root authentication token to the second client and the second client access the second space by sending to the second space one of the messages specified by the second schema. Ford and Lehman, both singly and in combination, fail to disclose sending an authentication token to a second client, as shown above, and Wyckoff does not teach anything at all regarding authentication token and clearly does not disclose one client sending an authentication token to a second client. A client sending an authentication token to another client does not make sense in the T Spaces system, as outlined above. Furthermore, Wyckoff also fails to disclose the second client access the second space by sending to the second space one of the messages specified by the second schema. As with Ford and Lehman, Wyckoff does not mention anything about a schema specifying messages usable to invoke functions of a space, and also fails to mention a second client sending such a message to a second space. Please refer to the remarks above regarding the rejection of claims 3 and 14 as they apply to claim 25 as well.

Claim 26:

Regarding claim 26, Lehman, Ford and Wyckoff, individually and in any combination, do not teach or suggest a client modifying a security policy of the second space, whereby the second space is configured to permit access to a second client, as the Examiner implies. The Examiner has failed to cite any particular portion in Wyckoff, Lehman, or Ford, but only states that Wyckoff teaches access controls that include security policies. However, any access controls taught by Lehman, Ford and Wyckoff fail to include a requesting client modifying a security policy of a space, whereby the second space is configured to permit access to a second client. The T Spaces system only

allows users to establish security policies by setting user and group permissions on a Tuplespace basis (e.g. Ford, column 5, lines 10-12). Wyckoff does not include any additional access control features that involve a requesting client modifying a security policy for a space, whereby the second space is configured to permit access to a second client. Please note that Appellants' arguments above regarding claims 4 and 15 also apply to claim 26 with equal force.

Claim 27:

Regarding claim 27, Ford in view of Lehman in further view of Wyckoff fails to disclose or suggest a second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service. Please note that Appellants' arguments above regarding claims 5 and 16 also apply to claim 27.

Claim 29:

In regards to claim 29, Ford in view of Lehman in further view of Wyckoff fails to teach or suggest that the first schema is expressed in a data representation language; and wherein the second schema is expressed in the data representation language. Please refer to Appellants' arguments above regarding claims 7 and 18 as they also apply to claim 29.

Claim 30:

Regarding claim 19, Ford in view of Lehman in further view of Wyckoff fails to teach or suggest that the data representation language comprises eXtensible Markup Language (XML). Please refer to Appellants' arguments above regarding claims 8 and 19 as they also apply to claim 30.

Claim 31:

Regarding claim 31, Ford in view of Lehman in further view of Wyckoff does not teach or suggest reading a service advertisement stored in the first space, wherein the service advertisement comprises information which is usable to access and execute a second service, as the Examiner contends. Furthermore, Lehman in view of Ford and Wyckoff also fails to teach using the information in the service advertisement to execute the second service; generating a set of results of the second service in response to executing the second service, or publishing the set of results of the second service in the second space. Please refer to Appellants' arguments above regarding claims 9 and 20 as they apply here with equal force.

Claim 32:

Regarding claim 32, Ford in view of Lehman in further view of Wyckoff does not teach or suggest wherein in accessing the first space, the first client is operable to access the first space at a first address to a storage facility; wherein in requesting creation of the second space, the first client is operable to request creation of a second address to the storage facility; and wherein in accessing the second space, the first client is operable to access the second space at the second address to the storage facility. Instead, Ford, Lehman and Wyckoff teach that a command or request to a T Spaces server is a method invocation sent to the T Spaces server. The T Spaces server then determines an appropriate handler for the operator(s) in the request (Ford, column 7, lines 8-30, and Lehman, column 5, line 35- column 6, line 3). Please note that Appellants' arguments above regarding claims 10 and 21 apply to claim 32 with equal force.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-33 was erroneous, and reversal of her decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of \$500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit

Account No. 501505/5181-67100/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'R. Kowert', with a long horizontal stroke extending to the right.

Robert C. Kowert
Reg. No. 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: January 26, 2005

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A method comprising:

accessing a first space, wherein the first space comprises a first network-addressable storage location, wherein information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space;

a requesting client requesting creation of a second space by sending to the first space one of the messages specified by the first schema;

creating the second space in response to the requesting client requesting creation of the second space, wherein the second space is initially configured to permit access only to the requesting client, wherein the second space comprises a second network-addressable storage location, wherein information usable to access the second space is provided in an advertisement for the second space, wherein the advertisement for the second space comprises a second schema, and wherein the second schema specifies one or more messages usable to invoke functions of the second space; and

the requesting client accessing the second space by sending to the second space one of the messages specified by the second schema.

2. The method of claim 1, further comprising:

creating a root authentication token for the second space;

initializing an authentication service associated with the second space, whereby the second space is configured to permit access only to a client holding the root authentication token; and

sending the root authentication token to the requesting client.

3. The method of claim 2, further comprising:

the requesting client sending the root authentication token to a second client; and

the second client accessing the second space by sending to the second space one of the messages specified by the second schema.

4. The method of claim 1, further comprising the requesting client modifying a security policy of the second space, whereby the second space is configured to permit access to a second client.

5. The method of claim 4, further comprising the second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service.

6. The method of claim 1,

wherein the accessing the first space comprises sending information to the first space at a first Uniform Resource Identifier (URI); and

wherein the requesting client accessing the second space comprises the requesting client sending information to the second space at a second URI.

7. The method of claim 1,

wherein the first schema is expressed in a data representation language; and

wherein the second schema is expressed in the data representation language.

8. The method of claim 7, wherein the data representation language comprises eXtensible Markup Language (XML).

9. The method of claim 1, further comprising:

reading a service advertisement stored in the first space, wherein the service advertisement comprises information which is usable to access and execute a second service;

using the information in the service advertisement to execute the second service;

generating a set of results of the second service in response to the executing the second service; and

publishing the set of results of the second service in the second space;

wherein the requesting creation of the second space comprises requesting creation of the second space for storage of the set of results of the service.

10. The method of claim 1,

wherein the accessing the first space comprises accessing the first space at a first address to a storage facility;

wherein the creating the second space comprises creating a second address to the storage facility; and

wherein the accessing the second space comprises accessing the second space at the second address to the storage facility.

11. The method of claim 1,

wherein the functions of the first space comprise storing information in the first space and reading information from the first space; and

wherein the functions of the second space comprise storing information in the second space and reading information from the second space.

12. A system comprising:

a first client;

a first space which is communicatively coupled to the client, wherein the first space comprises a first network-addressable storage location, wherein information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space;

wherein the first client is operable to:

access the first space;

request creation of a second space by sending to the first space one of the messages specified by the first schema, wherein the second space

is initially configured to permit access only to the first client, wherein the second space comprises a second network-addressable storage location, wherein information usable to access the second space is provided in an advertisement for the second space, wherein the advertisement for the second space comprises a second schema, and wherein the second schema specifies one or more messages usable to invoke functions of the second space; and

access the second space by sending to the second space one of the messages specified by the second schema.

13. The system of claim 12,

wherein the second space is configured to permit access only to a client holding a root authentication token; and

wherein the second space is operable to send the root authentication token to the first client.

14. The system of claim 13, further comprising:

a second client which is communicatively coupled to the first client and the second space;

wherein the first client is operable to send the root authentication token to the second client; and

wherein the second client is operable to access the second space by sending to the second space one of the messages specified by the second schema.

15. The system of claim 12, further comprising:

a second client which is communicatively coupled to the first client and the second space;

wherein the first client is operable to modify a security policy of the second space, whereby the second space is configured to permit access to the second client.

16. The system of claim 15, wherein the second client is operable to read a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service.

17. The system of claim 12,

wherein in accessing the first space, the first client is operable to send information to the first space at a first URI; and

wherein in accessing the second space, the first client is operable to send information to the second space at a second URI.

18. The system of claim 12,

wherein the first schema is expressed in a data representation language; and

wherein the second schema is expressed in the data representation language.

19. The system of claim 18, wherein the data representation language comprises eXtensible Markup Language (XML).

20. The system of claim 12, further comprising:

a service which is communicatively coupled to the first client and to the first space;

wherein the first client is operable to read a service advertisement stored in the first space, wherein the service advertisement comprises information which is usable to access and execute the service;

wherein the service is operable to:

generate a set of results of executing the service;

create the second space; and

publish the set of results in the second space.

21. The system of claim 12,

wherein in accessing the first space, the first client is operable to access the first space at a first address to a storage facility;

wherein in requesting creation of the second space, the first client is operable to request creation of a second address to the storage facility; and

wherein in accessing the second space, the first client is operable to access the second space at the second address to the storage facility.

22. The system of claim 12,

wherein the functions of the first space comprise storing information in the first space and reading information from the first space; and

wherein the functions of the second space comprise storing information in the second space and reading information from the second space.

23. A carrier medium comprising program instructions which are computer-executable to implement:

accessing a first space, wherein the first space comprises a first network-addressable storage location, wherein information usable to access the first space is provided in an advertisement for the first space, wherein the advertisement for the first space comprises a first schema, and wherein the first schema specifies one or more messages usable to invoke functions of the first space;

a requesting client requesting creation of a second space by sending to the first space one of the messages specified by the first schema;

creating the second space in response to the requesting client requesting creation of the second space, wherein the second space is initially configured to permit access only to the requesting client, wherein the second space comprises a second network-addressable storage location, wherein information usable to access the second space is provided in an advertisement for the second space, wherein the advertisement for the second space comprises a second schema, and wherein the second schema specifies one or more messages usable to invoke functions of the second space; and

the requesting client accessing the second space by sending to the second space one of the messages specified by the second schema.

24. The carrier medium of claim 23, wherein the program instructions are further computer-executable to implement:

creating a root authentication token for the second space;

initializing an authentication service associated with the second space, whereby the second space is configured to permit access only to a client holding the root authentication token; and

sending the root authentication token to the requesting client.

25. The carrier medium of claim 24, wherein the program instructions are further computer-executable to implement:

the requesting client sending the root authentication token to a second client; and

the second client accessing the second space by sending to the second space one of the messages specified by the second schema.

26. The carrier medium of claim 23, wherein the program instructions are further computer-executable to implement the requesting client modifying a security policy of the second space, whereby the second space is configured to permit access to a second client.

27. The carrier medium of claim 26, wherein the program instructions are further computer-executable to implement the second client reading a service advertisement from the second space, wherein the service advertisement comprises information usable to execute a corresponding service.

28. The carrier medium of claim 23,

wherein in the accessing the first space, the program instructions are further computer-executable to implement sending information to the first space at a first URI; and

wherein in the requesting client accessing the second space, the program instructions are further computer-executable to implement the requesting client sending information to the second space at a second URI.

29. The carrier medium of claim 23,

wherein the first schema is expressed in a data representation language; and

wherein the second schema is expressed in the data representation language.

30. The carrier medium of claim 29, wherein the data representation language comprises eXtensible Markup Language (XML).

31. The carrier medium of claim 23, wherein the program instructions are further computer-executable to implement:

reading a service advertisement stored in the first space, wherein the service advertisement comprises information which is usable to access and execute a second service;

using the information in the service advertisement to execute the second service;

generating a set of results of the second service in response to the executing the second service; and

publishing the set of results of the second service in the second space;

wherein in the requesting creation of the second space, the program instructions are further computer-executable to implement requesting creation of the second space for storage of the set of results of the second service.

32. The carrier medium of claim 23,

wherein in the accessing the first space, the program instructions are further computer-executable to implement accessing the first space at a first address to a storage facility;

wherein in the creating the second space, the program instructions are further computer-executable to implement creating a second address to the storage facility; and

wherein in the accessing the second space, the program instructions are further computer-executable to implement accessing the second space at the second address to the storage facility.

33. The carrier medium of claim 23,

wherein the functions of the first space comprise storing information in the first space and reading information from the first space; and

wherein the functions of the second space comprise storing information in the second space and reading information from the second space.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.